WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Thursday, January 03, 2008

Hide?	Set Name	Query	Hit Count
		DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=NO; OP=OR	
☐	L130	(l116 or l117 or l118 or l119 or l120 or l121 or l122 or l123 or l124 or L125) and l94	1
☐	L129	(l116 or l117 or l118 or l119 or l120 or l121 or l122 or l123 or l124 or L125) and l93	1
☐	L128	(l116 or l117 or l118 or l119 or l120 or l121 or l122 or l123 or l124 or L125) and l87	1
☐	L127	(l116 or l117 or l118 or l119 or l120 or l121 or l122 or l123 or l124 or L125) and l86	1
☐	L126	(l116 or l117 or l118 or l119 or l120 or l121 or l122 or l123 or l124 or L125) and (l81 or l82)	4
☐	L125	711/217.ccls.	523
☐	L124	711/216.ccls.	352
☐	L123	711/212.ccls.	240
☐	L122	711/145.ccls.	910
☐	L121	711/100.ccls.	1762
☐	L120	711/1.ccls.	587
☐	L119	707/206.ccls.	815
☐	L118	707/101.ccls.	3562
☐	L117	707/100.ccls.	5915
☐	L116	707/8.ccls.	1284
☐	L115	(L112 or L113 or L114) and (L106 or L107 or L108 or L109 or L110 or L111)	1
☐	L114	ZHOU-XIN.in.	56
☐	L113	LUEH-GUEI-YUAN.in.	60
☐	L112	WU-GANSHA.in.	37
		DB=PGPB,USPT,USOC; PLUR=NO; OP=OR	
☐	L111	L94 and lock$	1
☐	L110	L94 and (lock adj1 (information or data))	1
☐	L109	L93 and lock$	1
☐	L108	L93 and (lock adj1 (information or data))	1
☐	L107	L94 and (hash adj1 code)	1
☐	L106	L93 and (hash adj1 code)	1
		DB=EPAB,JPAB,DWPI,TDBD; PLUR=NO; OP=OR	

10\790,230

☐	L105	L101 and L104	1
☐	L104	"least significant bits"	1108
☐	L103	L101 and (encod$ near (bit or bits))	0
☐	L102	L101 and ((k adj1 bit) or k-bit)	0
☐	L101	(align$ near (address or addresses))	137
		DB=PGPB,USPT,USOC; PLUR=NO; OP=OR	
☐	L100	L98 and (garbage near collect$)	1
☐	L99	L97 and (garbage near collect$)	1
☐	L98	L96 and L85	56
☐	L97	L95 and L85	56
☐	L96	L82 and L83	56
☐	L95	L81 and L83	56
☐	L94	L91 and (garbage near collect$)	1
☐	L93	L92 and (garbage near collect$)	1
☐	L92	L85 and L89	56
☐	L91	L85 and L90	56
☐	L90	L87 and L84	56
☐	L89	L86 and L84	56
☐	L88	L86 and l19	0
☐	L87	L82 and L85	64
☐	L86	L81 and L85	56
☐	L85	((bit or bits) near zero)	20180
☐	L84	"least significant bits"	19809
☐	L83	(n near (bit or bits))	45767
☐	L82	L79 and (encod$ same (bit or bits))	71
☐	L81	L79 and (encod$ near (bit or bits))	63
☐	L80	L76 and L78	1
☐	L79	L77 and L78	79
☐	L78	((k adj1 bit) or k-bit)	5007
☐	L77	(align$ near (address or addresses))	2235
☐	L76	(align near (address or addresses))	133
☐	L75	(L68 or L69) and L74	6
☐	L74	L67 and L73	146
☐	L73	((encod$ or lock$ or hash$ or (garbage adj1 collect$)) near (bit or bits))	26932
☐	L72	L71 and address$	17
☐	L71	L70 and (bit or bits)	17
☐	L70	L69 and (word or words)	20

☐	L69	(object adj1 header).ab.	41
☐	L68	(object adj1 header).ti.	1
☐	L67	(object adj1 header)	1233
☐	L66	L64 and (encod$ near (bit or bits))	22
☐	L65	L64 and (encod$ near lock$)	1
☐	L64	(L58 or L59 or L60 or L61 or L62 or L63) and ((virtual address) with (bit or bits) with (object or objects or class or table or tables))	312
☐	L63	(707/103R |707/103Y |707/103X |707/103Z).ccls.	2187
☐	L62	(707/100).ccls.	5915
☐	L61	(707/8).ccls.	1284
☐	L60	(711/100).ccls.	1761
☐	L59	(711/6).ccls.	269
☐	L58	(711/2).ccls.	370
☐	L57	L56 and address.ti.	47
☐	L56	((bit or bits) near lock)	1984
☐	L55	(two adj1 bit adj1 lock)	5
☐	L54	(two-bit adj1 lock)	3
☐	L53	(two-bit lock)	792355
☐	L52	L49 and (lock near (bit or bits))	5
☐	L51	L49 and (lock with (bit or bits))	22
☐	L50	L49 and lock	99
☐	L49	L20 and address.ti.	2306
☐	L48	L47 and address.ti.	0
☐	L47	L20 and (lock with encode)	51
☐	L46	L45 and (word or words)	1
☐	L45	6987813.pn.	1
☐	L44	L43 and address.ti.	40
☐	L43	L20 and (address same word same n same (bit or bits))	1543
☐	L42	L41 and virtual	12
☐	L41	L40 and encod$	58
☐	L40	L38 and L39	232
☐	L39	address.ti.	7410
☐	L38	(address same word same n same (bit or bits))	7400
☐	L37	L36 and (n near (bit or bits))	0
☐	L36	(virtual with encode with (bit or bits) with address)	5
☐	L35	((object or objects) with virtual with encode with (bit or bits) with address)	0
☐	L34	L32 and memory	28
☐	L33	L32 an dmemory	18268

☐	L32	L31 and (pointer or pointers)	31
☐	L31	L30 and encod$	80
☐	L30	L28 and header.ab.	149
☐	L29	L28 and header.ti.	14
☐	L28	L20 and (address near (bit or bits))	11810
☐	L27	L25 and (address near k-bit)	1
☐	L26	L24 and L25	1
☐	L25	L20 and (encod$ near lock$)	96
☐	L24	L23 and (pointer or pointers)	2
☐	L23	L22 and lock	2
☐	L22	L21 and memory	18
☐	L21	L20 and (address near k-bit)	18
	DB=PGPB; PLUR=NO; OP=OR		
☐	L20	2 5551952 5597190 5633515 5666158 5804748 5850927 6012554 6043646 7071869 20060087473 6418140 6633592 5539871 6078931 6085196 4294543 4777978 5347629 5360214 5685776 5835591 5893798 6334212 6424354 20060082597 6098185 6278838 6341198 5600768 6324178 6334213 6442752 20020131408 20020167948 6907437 6138202 6135653 5002479 5467472 5680616 4432170 4498023 4992931 5339312 5430839 5432937 5495509 5604905 5610660 5617455 5729755 5754564 5768447 5819283 5818447 5852664 5974548 6175862 6199100 6249319 6249788 6317500 6351815 5649218 5834040 6026192 6359910 20040124158 20050115911 6147944 6400889 6173258 6475871 20060107175 5710499 6134627 5900001 5920876 5687368 5761510 5832268 5970242 6223344 6223344 6237043 4772882 4939672 5289536 5325492 5367573 5611076 5705762 5713014 5764987 5825767 5875299 5903900 5911144 5915255 5930807 5948058 5968157 5999730 6038572 6049810 6052729 6115782 6116008 6219830 6219830 6253252 6263338 6314558 6373846 6434685 6457019 6499137 6567837 6678697 6691304 6754898 6886159 6931423 7028287 20020099902 20020194191 20030093577 20030097396 20030105772 20030236819 20040001010 20040205701 20050149587 20060077993 20060126956 4458533 4467443 4580264 4841486 5278906 6064958 6964018 6999622 20010036318 20050182794 20050188296 20040252119 5993218 4927147 5727788 5839902 6324631 5471471 5471472 5657314 5673318 5757913 5844907 6389031 20030043806 4341521 4344243 4378118 4495230 4573927 4583006 5180337 5291674 5345961 5528496 5557202 5677626 6246231 20040200172 20060016139 5960436 5276868 5471487 5623449 5644784 5968125 6055616 6075931 6138208 6260011 20030225998 20050223329 20060106986 5201277 5878427 5893913 5999648 6078926 4290421 4364056 4420265 4511248 4608662 4755955 4756448 4774666 4910103 4931999 4951241 4951238 4958159 4984183 5195147 5208905 5311488 5321801 5361131 5392432 5483623 5485613 5505557 5574439 5696918 5752053 5763308 5787283 5812794 5832225 5841145 5897647 5944533 6070168 6201959 6240406 6259629 6438036 20050253335 3848710 3816709 4057342 4243319 4243270	1648918
	DB=PGPB,USPT,USOC; PLUR=NO; OP=OR		

☐	L19	L18 and (runtime or run-time)	0
☐	L18	6987813.pn.	1
☐	L17	L16 and (n near (bit or bits))	15
☐	L16	(bit adj1 stealing)	87
☐	L15	L4 and (address same word same n same (bit or bits))	33
☐	L14	L3 and L13	4
☐	L13	(address same word same n same (bit or bits))	7400
☐	L12	L11 and (address same n same (bit or bits))	9
☐	L11	L3 and (n near (bit or bits))	29
☐	L10	L9 and (n near (bit or bits))	2
☐	L9	L3 and (header with pointer with (bit or bits))	18
☐	L8	L6 and (header with pointer with (bit or bits))	9
☐	L7	L6 and (k near (bit or bits))	28
☐	L6	L5 and (n near (bit or bits))	153
☐	L5	L4 and bit$.ab.	956
☐	L4	header.ab.	13759
☐	L3	header.ti.	2641
☐	L2	(compact near object near header)	1
	DB=USPT; PLUR=NO; OP=OR		
☐	L1	6748402.pn.	1

END OF SEARCH HISTORY

PORTAL
USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ● The ACM Digital Library ○ The Guide

aligning address and k-bit word and ecoding and bits and garb SEARCH

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used:
aligning address and **k bit word** and **ecoding** and **bits** and **garbage collection**

Found **41,479** of **216,412**

Sort results by: relevance
Display results: expanded form

Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next
Best 200 shown
Relevance scale

1 Compiler construction: an advanced course
F. L. Bauer, F. L. De Remer, M. Griffiths, U. Hill, J. J. Horning, C. H. A. Koster, W. M. McKeeman, P. C. Poole, W. M. Waite, G. Goos, J. Hartmanis
January 1974 Book
Publisher: Springer-Verlag New York, Inc.
Additional Information: full citation, abstract, references, cited by

The Advanced Course took place from March 4 to 15, 1974 and was organized by the Mathematical Institute of the Technical University of Munich and the Leibniz Computing Center of the Bavarian Academy of Sciences, in co-operation with the European Communities, sponsored by the Ministry for Research and Technology of the Federal Republic of Germany and by the European Research Office, London.

2 Smalltalk-80: the language and its implementation
Adele Goldberg, David Robson
January 1983 Book
Publisher: Addison-Wesley Longman Publishing Co., Inc.
Full text available: pdf(33.56 MB) Additional Information: full citation, abstract, cited by, index terms, review

From the Preface (See Front Matter for full Preface)

Advances in the design and production of computer hardware have brought many more people into direct contact with computers. Similar advances in the design and production of computer software are required in order that this increased contact be as rewarding as possible. The Smalltalk-80 system is a result of a decade of research into creating computer software that is appropriate for producing highly functional and interactive ...

3 Efficient implementation of bit-vector operation in Common Lisp
Henry G. Baker
April 1990 **ACM SIGPLAN Lisp Pointers**, Volume III Issue 2-4
Publisher: ACM Press
Full text available: pdf(1.24 MB) Additional Information: full citation, abstract, index terms

In this paper we show various techniques for the efficient implementation of the various functions of Common Lisp involving bit-vectors and bit-arrays. Bit-vectors are extremely

10/790,230

useful for computing everything from the Sieve of Eratosthenes for finding prime numbers, to the representation of sets and relations, to the implementation of natural language parsers, to the performance of *flow analysis* in an optimizing compiler, to the manipulation of complex communication codes like those used ...

4 Programming languages: Garbage collection for embedded systems
David F. Bacon, Perry Cheng, David Grove
September 2004 **Proceedings of the 4th ACM international conference on Embedded software EMSOFT '04**
Publisher: ACM Press
Full text available: pdf(199.59 KB) Additional Information: full citation, abstract, references, citings, index terms

Security concerns on embedded devices like cellular phones make Java an extremely attractive technology for providing third-party and user-downloadable functionality. However, garbage collectors have typically required several times the maximum live data set size (which is the minimum possible heap size) in order to run well. In addition, the size of the virtual machine (ROM) image and the size of the collector's data structures (metadata) have not been a concern for server- or workstation-orien ...

Keywords: compaction, fragmentation, mark-and-sweep, tracing

5 Cost-effective object space management for hardware-assisted real-time garbage collection
Kelvin D. Nilsen, William J. Schmidt
December 1992 **ACM Letters on Programming Languages and Systems (LOPLAS)**, Volume 1 Issue 4
Publisher: ACM Press
Full text available: pdf(1.29 MB) Additional Information: full citation, abstract, references, citings, index terms

Modern object-oriented languages and programming paradigms require finer-grain division of memory than is provided by traditional paging and segmentation systems. This paper describes the design of an OSM (Object Space Manager) that allows partitioning of real memory on object, rather than page, boundaries. The time required by the OSM to create an object, or to find the beginning of an object given a pointer to any location within it, is approximately one memory cycle. Object sizes are lim ...

Keywords: automatic garbage collection, dynamic storage management, high-level language architectures, memory technologies, real-time and embedded systems, run-time environments

6 New garbage collection algorithms and strategies: Dynamic selection of application-specific garbage collectors
Sunil Soman, Chandra Krintz, David F. Bacon
October 2004 **Proceedings of the 4th international symposium on Memory management ISMM '04**
Publisher: ACM Press
Full text available: pdf(185.74 KB) Additional Information: full citation, abstract, references, citings, index terms

Much prior work has shown that the performance enabled by garbage collection (GC) systems is highly dependent upon the behavior of the application as well as on the available resources. That is, no single GC enables the best performance for all programs and all heap sizes. To address this limitation, we present the design, implementation, and empirical evaluation of a novel Java Virtual Machine (JVM) extension that facilitates dynamic switching between a number of very different and popular g ...

Keywords: Java, annotation, application-specific collection, dynamic selection, hot-swapping, virtual machine

7 Cycles to recycle: garbage collection to the IA-64
Richard L. Hudson, J. Elliot Moss, Sreenivas Subramoney, Weldon Washburn
October 2000 **ACM SIGPLAN Notices , Proceedings of the 2nd international symposium on Memory management ISMM '00**, Volume 36 Issue 1
Publisher: ACM Press
Full text available: pdf(1.25 MB) Additional Information: full citation, abstract, citings, index terms

The IA-64, Intel's 64-bit instruction set architecture, exhibits a number of interesting architectural features. Here we consider those features as they relate to supporting garbage collection (GC). We aim to assist GC and compiler implementors by describing how one may exploit features of the IA-64. Along the way, we record some previously unpublished object scanning techniques, and offer novel ones for object allocation (suggesting some simple operating system support that would simplify it ...

8 Tag-free garbage collection for strongly typed programming languages
Benjamin Goldberg
May 1991 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1991 conference on Programming language design and implementation PLDI '91**, Volume 26 Issue 6
Publisher: ACM Press
Full text available: pdf(1.03 MB) Additional Information: full citation, references, citings, index terms

9 Concurrent garbage collection using hardware-assisted profiling
Timothy H. Heil, James E. Smith
October 2000 **ACM SIGPLAN Notices , Proceedings of the 2nd international symposium on Memory management ISMM '00**, Volume 36 Issue 1
Publisher: ACM Press
Full text available: pdf(1.74 MB) Additional Information: full citation, abstract, citings, index terms

In the presence of on-chip multithreading, a Virtual Machine (VM) implementation can readily take advantage of *service threads* for enhancing performance by performing tasks such as profile collection and analysis, dynamic optimization, and garbage collection concurrently with program execution. In this context, a hardware-assisted profiling mechanism is proposed. The *Relational Profiling Architecture* (RPA) is designed from the top down. RPA is based on a relational model similar ...

10 Conservative garbage collection for general memory allocators
Gustavo Rodriguez-Rivera
October 2000 **ACM SIGPLAN Notices , Proceedings of the 2nd international symposium on Memory management ISMM '00**, Volume 36 Issue 1
Publisher: ACM Press
Full text available: pdf(829.20 KB) Additional Information: full citation, abstract, citings, index terms

This paper explains a technique that integrates conservative garbage collection on top of general memory allocators. This is possible by using two data structures named malloc-tables and jump-tables that are computed at garbage collection time to map pointers to beginning of objects and their sizes. This paper describes malloc-tables and jump-tables, an implementation of a malloc/jump-table based conservative garbage collector for Doug Lea's memory allocator, and experimental results that com ...

Keywords: automatic memory management, conservative garbage collection, memory allocation

11 Ada development system technical and performance requirements (with rationale)
Donald G. Krantz
December 1990 **Proceedings of the conference on TRI-ADA '90 TRI-Ada '90**
Publisher: ACM Press
Full text available: pdf(1.85 MB) Additional Information: full citation, abstract, references

This paper discusses requirements for Ada1 compilers and associated tools used for real-time embedded weapons systems (EWS) development. The requirements have been developed over a period of several years by embedded systems developers at Honeywell Inc. and Alliant Techsystems Inc. Requirements for the run time system, compiler-generated code, and host tools such as linkers are presented. A short rationale statement is provided with each specific requirement.

12 Age-based garbage collection
Darko Stefanović, Kathryn S. McKinley, J. Eliot B. Moss
October 1999 **ACM SIGPLAN Notices , Proceedings of the 14th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '99**, Volume 34 Issue 10
Publisher: ACM Press
Full text available: pdf(1.47 MB) Additional Information: full citation, abstract, references, citings, index terms

Modern generational garbage collectors look for garbage among the young objects, because they have high mortality; however, these objects include the very youngest objects, which clearly are still live. We introduce new garbage collection algorithms, called age-based, some of which postpone consideration of the youngest objects. Collecting less than the whole heap requires write barrier mechanisms to track pointers into the collected region. We describe her ...

Keywords: garbage collection, generational and copy collection, object behavior, write barrier

13 The measured cost of copying garbage collection mechanisms
Michael W. Hicks, Jonathan T. Moore, Scott M. Nettles
August 1997 **ACM SIGPLAN Notices , Proceedings of the second ACM SIGPLAN international conference on Functional programming ICFP '97**, Volume 32 Issue 8
Publisher: ACM Press
Full text available: pdf(1.65 MB) Additional Information: full citation, abstract, references, citings, index terms

We examine the costs and benefits of a variety of copying garbage collection (GC) mechanisms across multiple architectures and programming languages. Our study covers both low-level object representation and copying issues as well as the mechanisms needed to support more advanced techniques such as generational collection, large object spaces, and type segregated areas.Our experiments are made possible by a novel performance analysis tool, *Oscar*. Oscar allows us to capture snapshots of pr ...

14 MC^2: high-performance garbage collection for memory-constrained environments
Narendran Sachindran, J. Eliot B. Moss, Emery D. Berger
October 2004 **ACM SIGPLAN Notices , Proceedings of the 19th annual ACM SIGPLAN**

conference on Object-oriented programming, systems, languages, and applications OOPSLA '04, Volume 39 Issue 10
Publisher: ACM Press
Full text available: pdf(503.53 KB) Additional Information: full citation, abstract, references, citings, index terms

Java is becoming an important platform for memory-constrained consumer devices such as PDAs and cellular phones, because it provides safety and portability. Since Java uses garbage collection, efficient garbage collectors that run in constrained memory are essential. Typical collection techniques used on these devices are mark-sweep and mark-compact. Mark-sweep collectors can provide good throughput and pause times but suffer from fragmentation. Mark-compact collectors prevent fragmentation, ...

Keywords: copying collector, generational collector, java, mark-compact, mark-copy, mark-sweep, memory-constrained copying

15 Compact garbage collection tables
David Tarditi
October 2000 **ACM SIGPLAN Notices , Proceedings of the 2nd international symposium on Memory management ISMM '00**, Volume 36 Issue 1
Publisher: ACM Press
Full text available: pdf(958.92 KB) Additional Information: full citation, abstract, index terms

Garbage collection tables for finding pointers on the stack can be represented in 20-25% of the space previously reported. Live pointer information is often the same at many call sites because there are few pointers live across most call sites. This allows live pointer information to be represented compactly by a small index into a table of descriptions of pointer locations. The mapping from program counter values to those small indexes can be represented compactly using several techniques. T ...

16 Space efficient conservative garbage collection
Hans-Juergen Boehm
June 1993 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1993 conference on Programming language design and implementation PLDI '93**, Volume 28 Issue 6
Publisher: ACM Press
Full text available: pdf(1.03 MB) Additional Information: full citation, abstract, references, citings, index terms

We call a garbage collector conservative if it has only partial information about the location of pointers, and is thus forced to treat arbitrary bit patterns as though they might be pointers, in at least some cases. We show that some very inexpensive, but previously unused techniques can have dramatic impact on the effectiveness of conservative garbage collectors in reclaiming memory. Our most significant observation is that static data that appears to point to the heap should not result i ...

17 Selected writings on computing: a personal perspective
Edsger W. Dijkstra
January 1982 Book
Publisher: Springer-Verlag New York, Inc.
Additional Information: full citation, abstract, references, cited by, index terms

Since the summer of 1973, when I became a Burroughs Research Fellow, my life has been very different from what it had been before. The daily routine changed: instead of going to the University each day, where I used to spend most of my time in the company of others, I now went there only one day a week and was most of the time that is, when not travelling!-- alone in my study. In my solitude, mail and the written word in general became more and more important. The circumstance that my employe ...

18 An on-the-fly reference-counting garbage collector for java
Yossi Levanoni, Erez Petrank
January 2006 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 28 Issue 1
Publisher: ACM Press
Full text available: pdf(787.15 KB) Additional Information: full citation, abstract, references, index terms

Reference-counting is traditionally considered unsuitable for multiprocessor systems. According to conventional wisdom, the update of reference slots and reference-counts requires atomic or synchronized operations. In this work we demonstrate this is not the case by presenting a novel reference-counting algorithm suitable for a multiprocessor system that does not require any synchronized operation in its write barrier (not even a compare-and-swap type of synchronization). A second novelty of thi ...

Keywords: Programming languages, garbage collection, memory management, reference-counting

19 Mondrian memory protection
Emmett Witchel, Josh Cates, Krste Asanović
October 2002 **ACM SIGPLAN Notices , ACM SIGARCH Computer Architecture News , ACM SIGOPS Operating Systems Review , Proceedings of the 10th international conference on Architectural support for programming languages and operating systems ASPLOS-X**, Volume 37 , 30 , 36 Issue 10 , 5 , 5
Publisher: ACM
Full text available: pdf(1.53 MB) Additional Information: full citation, abstract, references, cited by

Mondrian memory protection (MMP) is a fine-grained protection scheme that allows multiple protection domains to flexibly share memory and export protected services. In contrast to earlier page-based systems, MMP allows arbitrary permissions control at the granularity of individual words. We use a compressed permissions table to reduce space overheads and employ two levels of permissions caching to reduce run-time overheads. The protection tables in our implementation add less than 9% overhead to ...

20 Java object header elimination for reduced memory consumption in 64-bit virtual machines
Kris Venstermans, Lieven Eeckhout, Koen De Bosschere
September 2007 **ACM Transactions on Architecture and Code Optimization (TACO)**, Volume 4 Issue 3
Publisher: ACM Press
Full text available: pdf(722.38 KB) Additional Information: full citation, abstract, references, index terms

Memory performance is an important design issue for contemporary computer systems given the huge processor/memory speed gap. This paper proposes a space-efficient Java object model for reducing the memory consumption of 64-bit Java virtual machines. We completely eliminate the object header through typed virtual addressing (TVA) or implicit typing. TVA encodes the object type in the object's virtual address by allocating all objects of a given type in a contiguous memory segment. This allows ...

Keywords: 64-bit implementation, Java object model, Virtual machine, implicit typing, typed virtual addressing

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Home | Login | Logout | Access Information | Alerts | Purchase History |

Welcome United States Patent and Trademark Office

☐ **Search Results**	**BROWSE**	**SEARCH**	**IEEE XPLORE GUIDE**

Results for **"(((object)<in>metadata) <and> ((header)<in>metadata))<and> ((word)<in..."**
Your search matched **3** of **1715275** documents.
A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

New [Beta] **Application Notes** POWERED BY SPEC

» **Search Options**

View Session History

New Search

» **Key**

IEEE JNL	IEEE Journal or Magazine
IET JNL	IET Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IET CNF	IET Conference Proceeding
IEEE STD	IEEE Standard

Modify Search

(((object)<in>metadata) <and> ((header)<in>metadata))<and> ((word)<in>metadat   Search

☐ Check to search only within this results set

Display Format: [✓ Citation] Citation & Abstract

IEEE/IET	**Books**	**Educational Courses**	**A**

IEEE/IET journals, transactions, letters, magazines, conference proceedings, and

view selected items	**Select All** **Deselect All**

☐ 1. **Space-efficient 64-bit Java objects through selective typed virtual addre**
Venstermans, K.; Eeckhout, L.; De Bosschere, K.;
Code Generation and Optimization, 2006. CGO 2006. International Symposiu
26-29 March 2006 Page(s):11 pp.
Digital Object Identifier 10.1109/CGO.2006.34
AbstractPlus | Full Text: PDF(416 KB)   IEEE CNF
Rights and Permissions

☐ 2. **Recognition of common areas in a Web page using visual information: a page classification**
Kovacevic, M.; Diligenti, M.; Gori, M.; Milutinovic, V.;
Data Mining, 2002. ICDM 2002. Proceedings. 2002 IEEE International Confer
9-12 Dec. 2002 Page(s):250.- 257
Digital Object Identifier 10.1109/ICDM.2002.1183910
AbstractPlus | Full Text: PDF(543 KB)   IEEE CNF
Rights and Permissions

☐ 3. **Name block location in facsimile images using spatial/visual cues**
Likforman-Sulem, L.;
Document Analysis and Recognition, 2001. Proceedings. Sixth International C
10-13 Sept. 2001 Page(s):680 - 684
Digital Object Identifier 10.1109/ICDAR.2001.953876
AbstractPlus | Full Text: PDF(336 KB)   IEEE CNF
Rights and Permissions

Help   Contact Us

Indexed by Inspec

10\ 790230

IEEE Xplore® RELEASE 2.4

Home | Login | Logout | Access Information | Alerts | Purchase History |

Welcome United States Patent and Trademark Office

☐ Search Results

BROWSE SEARCH IEEE XPLORE GUIDE

Results for **"(((k-bit)<in>metadata) <and> ((encode)<in>metadata))"**
Your search matched **8** of **1715275** documents.
A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

New [Beta] Application Notes POWERED BY GLOBALSPEC

Modify Search

(((k-bit)<in>metadata) <and> ((encode)<in>metadata)) Search

☐ Check to search only within this results set

Display Format: [✓ Citation] Citation & Abstract

» Search Options

View Session History

New Search

» Key

IEEE JNL	IEEE Journal or Magazine
IET JNL	IET Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IET CNF	IET Conference Proceeding
IEEE STD	IEEE Standard

IEEE/IET **Books** **Educational Courses** **A**

IEEE/IET journals, transactions, letters, magazines, conference proceedings, and

view selected items **Select All** **Deselect All**

☐ 1. **Bit-plane and pass dual parallel architecture for coefficient bit modeling**
Chao Xu; Yanju Han; Yizhen Zhang;
Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04)
Conference on
Volume 5, 17-21 May 2004 Page(s):V - 85-8 vol.5
Digital Object Identifier 10.1109/ICASSP.2004.1327053
AbstractPlus | Full Text: PDF(214 KB) **IEEE CNF**
Rights and Permissions

☐ 2. **On reducing transitions through data modifications**
Murgai, R.; Fujita, M.;
Design, Automation and Test in Europe Conference and Exhibition 1999. Pro
9-12 March 1999 Page(s):82 - 88
Digital Object Identifier 10.1109/DATE.1999.761101
AbstractPlus | Full Text: PDF(220 KB) **IEEE CNF**
Rights and Permissions

☐ 3. **Generalized Snake-in-the-Box Codes**
Singleton, R.C.;

Volume EC-15, Issue 4, Aug. 1966 Page(s):596 - 602
Digital Object Identifier 10.1109/PGEC.1966.264381
AbstractPlus | Full Text: PDF(1219 KB) **IEEE JNL**
Rights and Permissions

☐ 4. **Single chip MPEG audio decoder**
Maturi, G.;
Consumer Electronics, IEEE Transactions on
Volume 38, Issue 3, Aug 1992 Page(s):348 - 356
Digital Object Identifier 10.1109/30.156706
AbstractPlus | Full Text: PDF(436 KB) **IEEE JNL**
Rights and Permissions

5. **New motion estimation algorithm using adaptively quantized low bit-res**

☐ **VLSI architecture for MPEG2 video encoding**
Seongsoo Lee; Jeong-Min Kim; Suo-Ik Chae;
Circuits and Systems for Video Technology, IEEE Transactions on
Volume 8, Issue 6, Oct. 1998 Page(s):734 - 744
Digital Object Identifier 10.1109/76.728416
AbstractPlus | References | Full Text: PDF(300 KB) **IEEE JNL**
Rights and Permissions

☐ 6. **32-Parallel SAD Tree Hardwired Engine for Variable Block Size Motion E Real-Time Encoding Application**
Liu, Zhenyu; Song, Yang; Shao, Ming; Li, Shen; Li, Lingfeng; Goto, Satoshi; Il
Signal Processing Systems, 2007 IEEE Workshop on
17-19 Oct. 2007 Page(s):675 - 680
Digital Object Identifier 10.1109/SIPS.2007.4387630
AbstractPlus | Full Text: PDF(6077 KB) **IEEE CNF**
Rights and Permissions

☐ 7. **Fast Decoding of Fibonacci Encoded Texts**
Klein, S.T.;
Data Compression Conference, 2007. DCC '07
27-29 March 2007 Page(s):388 - 388
Digital Object Identifier 10.1109/DCC.2007.38
AbstractPlus | Full Text: PDF(68 KB) **IEEE CNF**
Rights and Permissions

☐ 8. **Optimal mapping for 2-bit high speed Huffman compression**
Bassiouni, M.A.; Mukherjee, A.;
Global Telecommunications Conference, 1993, including a Communications
Technical Program Conference Record, IEEE in Houston. GLOBECOM '93., I
29 Nov.-2 Dec. 1993 Page(s):1979 - 1983 vol.3
Digital Object Identifier 10.1109/GLOCOM.1993.318410
AbstractPlus | Full Text: PDF(404 KB) **IEEE CNF**
Rights and Permissions

Help Contact Us

Web Images Maps News Shopping Gmail more ▾   Sign in

Google

k-bit word and encoding and address and garb | Search | Advanced Search Preferences

The **"AND"** operator is unnecessary -- we include all search terms by default. [details]

Web Results **1** - **10** of about **188** for **k-bit word and encoding and address and garbage collection filetype:p**

An 8-**kbit** content-addressable and reentrant memory
where the on-chip **garbage collection** or storing the data is accomplished,. conditionally. ....
4) After the **address encoding** is over at the **word** "N", ...
ieeexplore.ieee.org/iel5/4/22604/01052420.pdf?arnumber=1052420 - Similar pages

On fault modeling and testing of content-addressable memories ...
The fourth function is the **garbage collection** function. which indicates whether the **word** location is ... without any **address** handling for **garbage word** loca- ...
ieeexplore.ieee.org/iel2/3170/8991/00397193.pdf?arnumber=39719 - Similar pages
[More results from ieeexplore.ieee.org]

[PDF] Microsoft PowerPoint - 3-concepts
File Format: PDF/Adobe Acrobat - View as HTML
Start with **k-bit** data **word**. • Add r check bits ...... This process is sometimes known as **garbage collection**. ECE 254 / CPS 225. 83. (C) 2006 Daniel J. Sorin ...
www.ee.duke.edu/~sorin/ece254/lectures/3-concepts.pdf - Similar pages

[PDF] Fully parallel integrated CAM/RAM using preclassification to ...
File Format: PDF/Adobe Acrobat - View as HTML
"**address**" is the encoded location of the matched **word**; and "match" is ..... means of "**garbage collection**" is employed, data may become ...
www.stanford.edu/class/ee371/handouts/ClassifyCam96.pdf - Similar pages

[PDF] Symbolic Boolean Manipulation with Ordered Binary Decision ...
File Format: PDF/Adobe Acrobat - View as HTML
Part of **garbage collection**. Move each variable through ordering to find its best .....
Represent as Boolean function. Over variables **encoding** states ...
www.ima.umn.edu/talks/workshops/aug2-13.99/bryant/bryant.pdf - Similar pages

[PDF] The C-- Language Specification Version 2.0 (CVS Revision 1.98)
File Format: PDF/Adobe Acrobat - View as HTML
compiler text) as well as a small run-time system with **garbage collector**. ... loads a 32-bit **word** from the memory location whose **address** is in the variable ...
www.cminusminus.org/extern/hman2.pdf - Similar pages

[PDF] show service-policy through show xlate Commands
File Format: PDF/Adobe Acrobat - View as HTML
This example shows the GTP statistics with the **word** gsn in the output. ...... **garbage collection** initiated connection closure ...
www.cisco.com/en/US/docs/security/fwsm/fwsm31/command/reference/s7.pdf - Similar pages

[PDF] Making Data Structures Confluently Persistent
File Format: PDF/Adobe Acrobat - View as HTML
in the fully persistent setting. Since each **address** of the naive scheme consists of O(1+. e (D). log U.) **words** it follows ...
www.math.tau.ac.il/~haimk/papers/journal2.pdf - Similar pages

10/790,230

Data structure and algorithms for new hardware technology
Write the **address** of the. section to the link part of the original data section. These writes are per-. formed by overwrite operations. **Garbage collection** ...
www.springerlink.com/index/f21u845g87v61736.pdf - Similar pages

[PDF] TurboChannel T1 Adapter
File Format: PDF/Adobe Acrobat - View as HTML
the next cycle the system module reads or writes a **word**; if the option module is not ready, it can ...... "Compacting **Garbage Collection** with Ambiguous ...
www.hpl.hp.com/techreports/Compaq-DEC/WRL-91-4.pdf - Similar pages

1 2 3 4 5 6 7 8 9 10 **Next**

Download Google Pack: free essential software for your PC

k-bit word and encoding and addres: Search

Search within results | Language Tools | Search Tips | Dissatisfied? Help us improve | Try Google Experimental